**Capstone Project Details**

Project 1: https://bstackdemo.com/

Project 2: <https://www.saucedemo.com/>

Project 3: <https://demo.nopcommerce.com/>

Project 4: <https://demowebshop.tricentis.com/>

Expecting to cover more than 40 Test steps for each URL

### **Tools & Tech Stack**

* **Language**: Java
* **Build Tool**: Maven (dependency management)
* **Test Frameworks**:
  + **TestNG** (parallel execution, grouping, configuration)
  + **Cucumber** (BDD style scenarios with Gherkin)
* **Reporting**: Extent Reports (rich HTML reports)
* **Version Control**: GitHub
* **Project Management**: JIRA (Agile stories, tasks, bug tracking)

### **Objective**

The objective of this capstone project is to design and implement a scalable, maintainable, and reusable **test automation framework** using **Java, Maven, TestNG, and Cucumber** that enables efficient **web application testing** across multiple demo websites. The framework will support **parallel execution, BDD-style test scenarios, and detailed reporting with Extent Reports**, while integrating with **GitHub for version control**, **JIRA for project management**, and optionally **Jenkins for CI/CD pipelines**. This project aims to simulate an industry-standard automation setup that improves **test coverage, traceability, and collaboration** within Agile teams.

### **Capstone Project Plan**

### **Team Organization**

### **7 Groups (6–7 members each), Team should collaborate and complete the project in Agile Model**

### **Roles per Group:**

### **Product Owner (PO): Defines scope & priorities**

### **Scrum Master (SM): Ensures Agile process & removes blockers**

### **Developers: Build automation scripts & maintain framework**

### **Testers: Design manual test cases, execute & validate automation**

### **Project Flow**

### **1. Requirement Analysis & Test Design (Create Manual Test Cases)**

### **Review application under test (AUT) functionalities (e.g., login, checkout, search, cart).**

### **Identify critical user journeys.**

### **Write Manual Test Cases in Excel/Jira (covering positive, negative, edge cases).**

### **Get review & sign-off from PO/SM.**

### **2. Project Setup**

### **Create a GitHub repository for each group/project.**

### **Clone repo locally → initialize Maven project structure.**

### **Configure dependencies (Selenium, TestNG, Cucumber, Extent Reports).**

### **Commit initial framework setup → push to GitHub.**

### **3. Selenium Automation**

### **Convert manual test cases → TestNG & BDD feature files (Cucumber).**

### **Implement step definitions & page object model classes in Java.**

**Use TestNG for execution control (parallel, grouping).**

**Use Excel & JDBC Integration and Extent Reporting**

**Validate functionality → push working automation scripts to GitHub.**

### **4. Jira Integration**

### **Create Epics → User Stories → Tasks → Subtasks.**

### **Link test cases to stories for traceability.**

### **Log defects when automated tests fail.**

### **Track sprint progress in Agile boards.**

### **5.Build project (Maven clean install)**

### **Run Selenium + TestNG + Cucumber tests**

### **Generate Extent HTML Reports**

### **6. Reporting & Analysis**

### **Generate Extent Reports after each test execution.**

### **Share execution summary with team via Jira/GitHub Wiki.**

### **Analyse failed tests → log defects in Jira.**

### **Refactor/reuse automation scripts for better stability & maintainability.**

### **7. Continuous Improvement**

### **Update manual & automated test cases based on new requirements/bugs.**

### **Conduct retrospectives at the end of each sprint.**

### **Enhance framework with utilities (e.g., reusable methods, listeners, hooks).**

### 

**---------------------**